Assignment #2

CSCI 581, Spring 2022

Jayaa Emekar

Instructions

In this assignment, you will be using the *ACME Sales* dataset to generate visualizations. Each question below asks you to provide a code section that will generate the requested chart type.

For full credit, make sure that each chart you generate

- has an appropriate title that clearly describes the information presented
- uses relevant labels for each axis
- deviates from any default style setting by changing the default (line/bar) color, marker shape, marker color, line width, or other feature
- uses any additional chart feature that will enhance the information your visualization is attempting to convey.

You are free to choose using `matplotlib` or `seaborn` or any other library to generate your visualization.

To get you started, the following code loads the *ACME Sales* dataset into a `pandas` `DataFrame` object and uses *Google Colab*'s interactive table feature to show you what the data looks like:

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
from google.colab import data_table

# Load the required dataset into a dataframe, df
df = pd.read_csv('https://www.ecst.csuchico.edu/~bjuliano/csci581/datasets/acme_sales.csv')

# Set up to use Google Colab's interactive tables feature
data_table.enable_dataframe_formatter()
```

```
# Display the interactive table
df
```

Out[ ]:

	month_number	facecream	facewash	toothpaste	bathsoap	shampoo	moisturizer	total_units	total_profit
0	1	2500	1500	5200	9200	1200	1500	21100	211000
1	2	2630	1200	5100	6100	2100	1200	18330	183300
2	3	2140	1340	4550	9550	3550	1340	22470	224700
3	4	3400	1130	5870	8870	1870	1130	22270	222700
4	5	3600	1740	4560	7760	1560	1740	20960	209600
5	6	2760	1555	4890	7490	1890	1555	20140	201400
6	7	2980	1120	4780	8980	1780	1120	29550	295500
7	8	3700	1400	5860	9960	2860	1400	36140	361400
8	9	3540	1780	6100	8100	2100	1780	23400	234000
9	10	1990	1890	8300	10300	2300	1890	26670	266700
10	11	2340	2100	7300	13300	2400	2100	41280	412800
11	12	2900	1760	7400	14400	1800	1760	30020	300200

1. Generate a line chart showing the company's total profit per month.

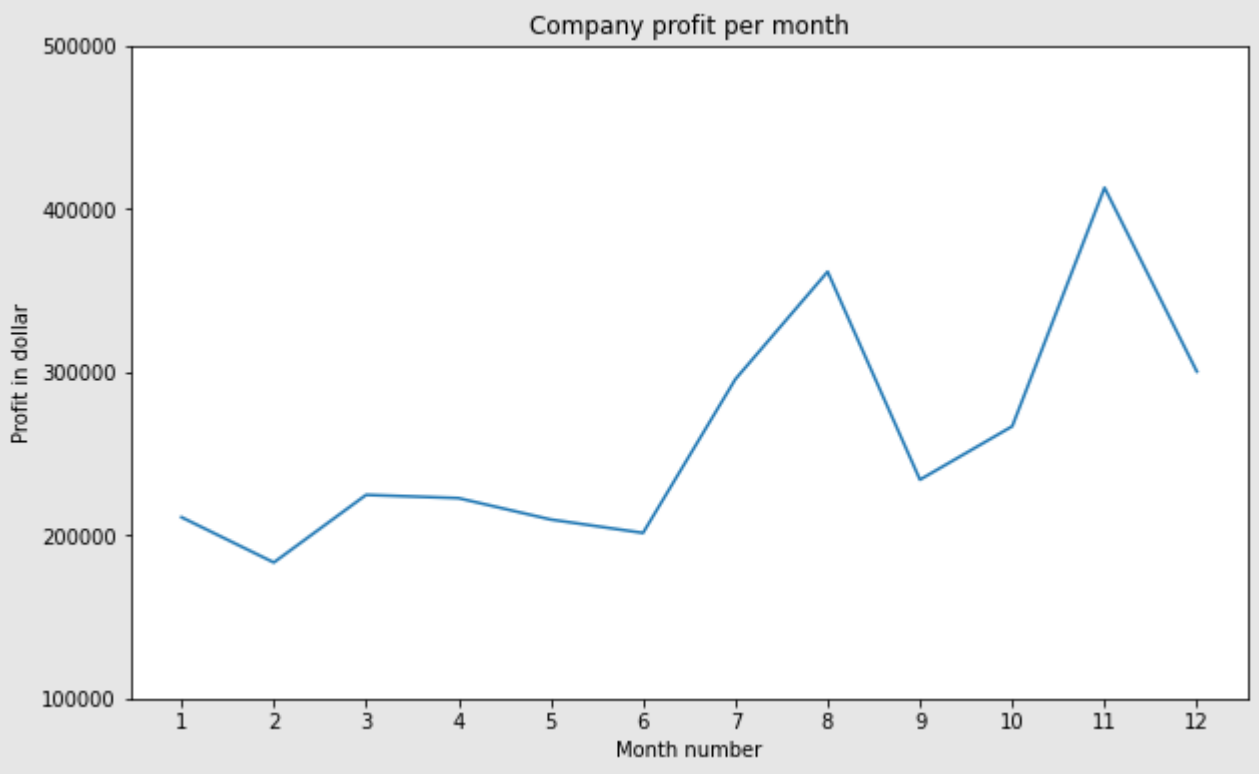
Solution:

In [ ]:

```
#Generate a line chart showing the company's total profit per month.
#Get data for month and profit
profitList = df ['total_profit'].tolist()
monthList  = df ['month_number'].tolist()

#Plot data for month and profit
plt.figure(figsize=(10, 6), facecolor='0.9')
plt.plot(monthList, profitList, label = 'Month-wise Profit data of last year')
```

```
plt.xlabel('Month number')
plt.ylabel('Profit in dollar')
plt.xticks(monthList)
plt.title('Company profit per month')
plt.yticks([100000, 200000, 300000, 400000, 500000])
plt.show()
```



2. Generate a multiline plot (*i.e.*, use a separate plot line for each of the six products) showing the total units sold per month for each product.

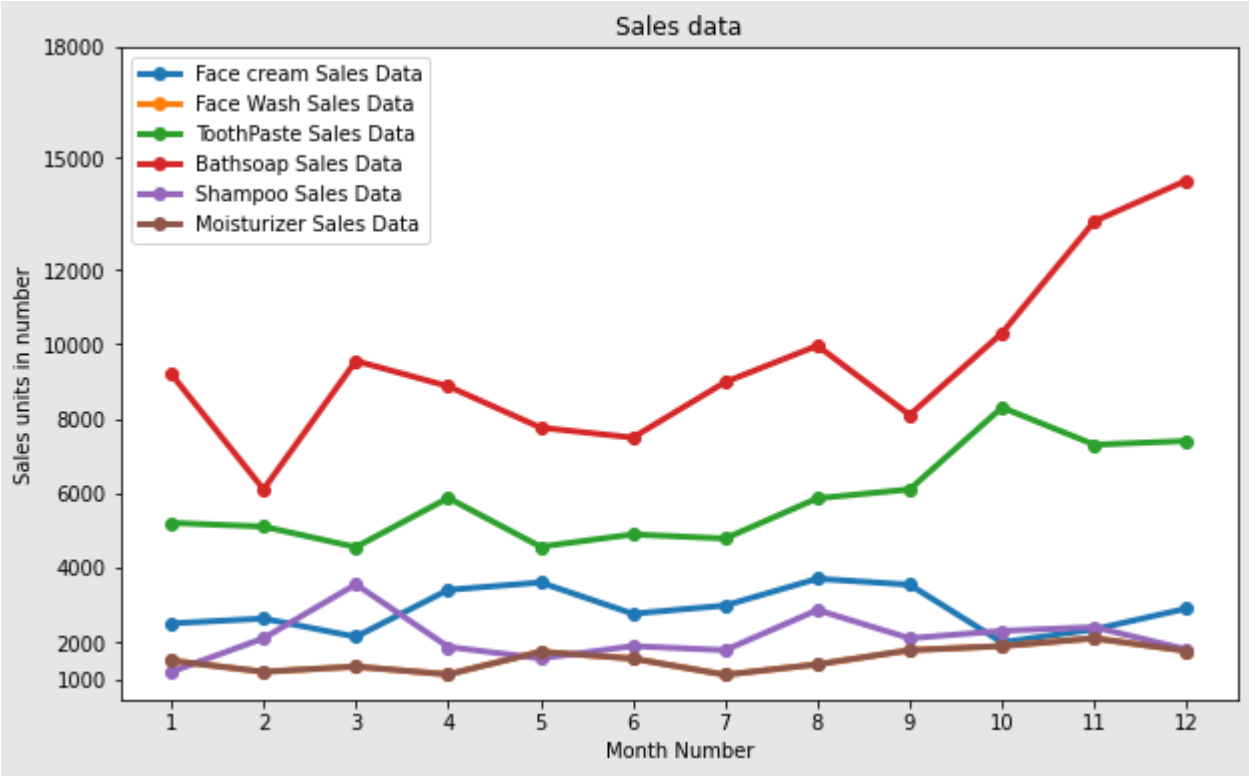
Solution(s)

In [ ]:
```
#Generate a multiline plot (i.e., use a separate plot line for each of the six products) showing the total units sold p
#Get data into variables
```

```
monthList  = df ['month_number'].tolist()
faceCremSalesData   = df ['facecream'].tolist()
faceWashSalesData   = df ['facewash'].tolist()
toothPasteSalesData = df ['toothpaste'].tolist()
bathingsoapSalesData   = df ['bathsoap'].tolist()
shampooSalesData   = df ['shampoo'].tolist()
moisturizerSalesData = df ['moisturizer'].tolist()

# Plot the multiline plot
plt.figure(figsize=(10, 6), facecolor='0.9')
plt.plot(monthList, faceCremSalesData,   label = 'Face cream Sales Data', marker='o', linewidth=3)
plt.plot(monthList, faceWashSalesData,   label = 'Face Wash Sales Data',  marker='o', linewidth=3)
plt.plot(monthList, toothPasteSalesData, label = 'ToothPaste Sales Data', marker='o', linewidth=3)
plt.plot(monthList, bathingsoapSalesData, label = 'Bathsoap Sales Data', marker='o', linewidth=3)
plt.plot(monthList, shampooSalesData, label = 'Shampoo Sales Data', marker='o', linewidth=3)
plt.plot(monthList, moisturizerSalesData, label = 'Moisturizer Sales Data', marker='o', linewidth=3)


plt.xlabel('Month Number')
plt.ylabel('Sales units in number')
plt.legend(loc='upper left')
plt.xticks(monthList)
plt.yticks([1000, 2000, 4000, 6000, 8000, 10000, 12000, 15000, 18000])
plt.title('Sales data')
plt.show()
```



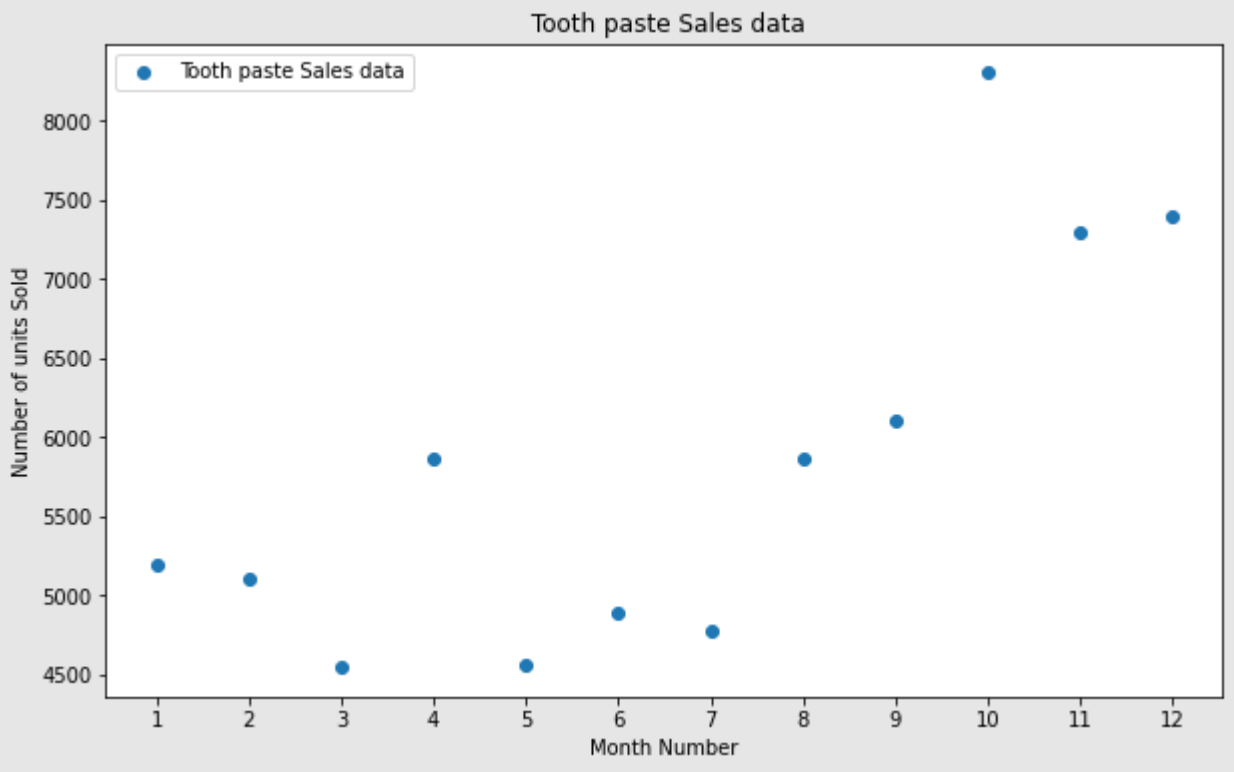
3. Generate a scatter plot that shows the toothpaste sales data for each month.

Solution(s)

In [ ]:
```
#Generate a scatter plot that shows the toothpaste sales data for each month.
monthList  = df ['month_number'].tolist()
toothPasteSalesData = df ['toothpaste'].tolist()

#plot the scatter plot
plt.figure(figsize=(10, 6), facecolor='0.9')
plt.scatter(monthList, toothPasteSalesData, label = 'Tooth paste Sales data')
plt.xlabel('Month Number')
plt.ylabel('Number of units Sold')
plt.legend(loc='upper left')
```

```
plt.title(' Tooth paste Sales data')
plt.xticks(monthList)
plt.show()
```



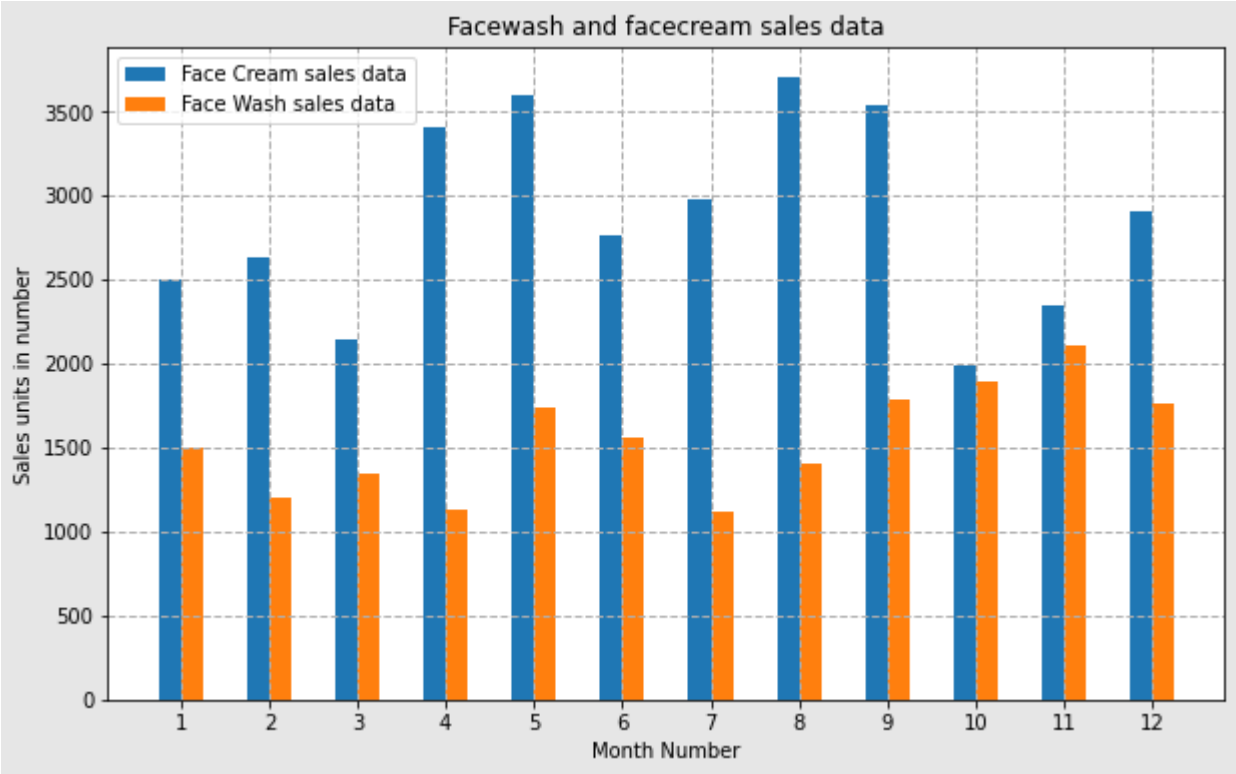
4. Generate a bar chart showing the face cream and face wash sales per month. Use a separate bar for each product in the same chart.

Solution(s)

In [ ]:
```
# gather the data from dataset
monthList  = df ['month_number'].tolist()
faceCremSalesData   = df ['facecream'].tolist()
faceWashSalesData   = df ['facewash'].tolist()
```

```
#plot the Facewash and facecream sales data
plt.figure(figsize=(10, 6), facecolor='0.9')
plt.bar([a-0.25 for a in monthList], faceCremSalesData, width= 0.25, label = 'Face Cream sales data', align='edge')
plt.bar([a+0.25 for a in monthList], faceWashSalesData, width= -0.25, label = 'Face Wash sales data', align='edge')
plt.xlabel('Month Number')
plt.ylabel('Sales units in number')
plt.legend(loc='upper left')
plt.title(' Sales data')

plt.xticks(monthList)
plt.grid(True, linewidth= 1, linestyle="--")
plt.title('Facewash and facecream sales data')
plt.show()
```

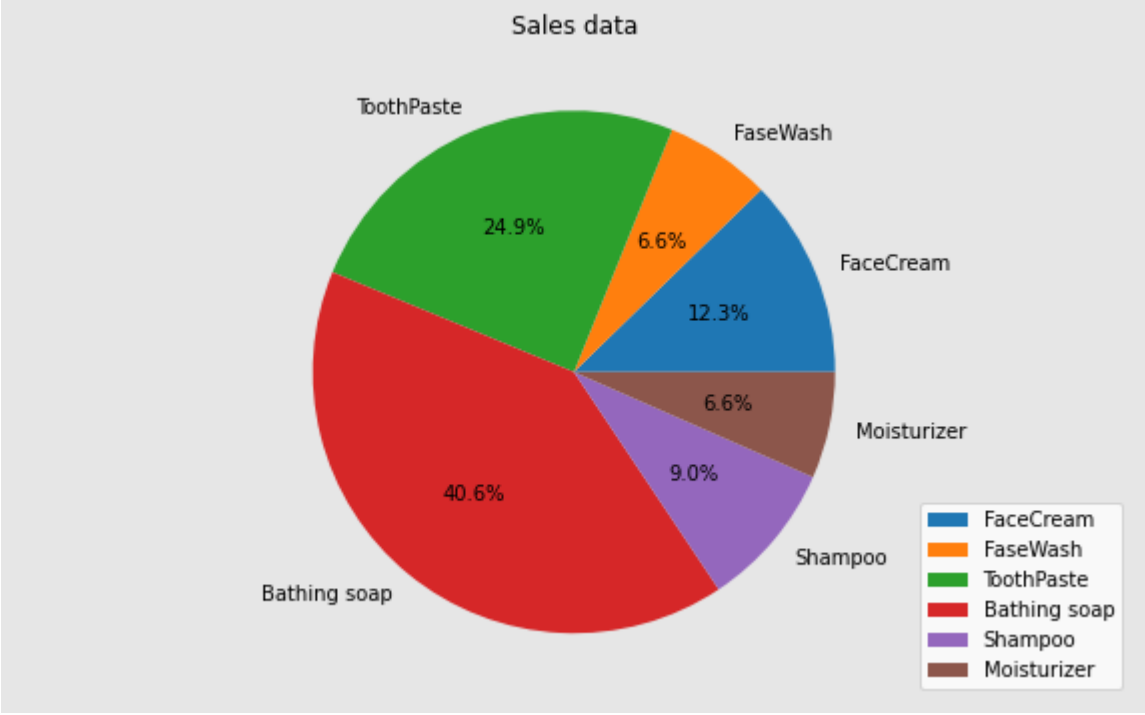


5. Generate a pie chart showing the sales proportion of each product for the total sale for the year. Explode the slice that has the highest proportion.

Solution(s)

In [ ]:
```
#Generate a pie chart showing the sales proportion of each product for the total sale for the year.
#Explode the slice that has the highest proportion.
monthList  = df ['month_number'].tolist()

plt.figure(figsize=(10, 6), facecolor='0.9')
labels = ['FaceCream', 'FaseWash', 'ToothPaste', 'Bathing soap', 'Shampoo', 'Moisturizer']
salesData   = [df ['facecream'].sum(), df ['facewash'].sum(), df ['toothpaste'].sum(),
         df ['bathsoap'].sum(), df ['shampoo'].sum(), df ['moisturizer'].sum()]
plt.axis("equal")
plt.pie(salesData, labels=labels, autopct='%1.1f%%')
plt.legend(loc='lower right')
plt.title('Sales data')
plt.show()
```



Bathing soap has highest contribution which is equal to 40.6%

Notes

(Optional) **Include any final thoughts, comments, or observations here, if applicable.**